MyFirstRMarkdownDocument

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

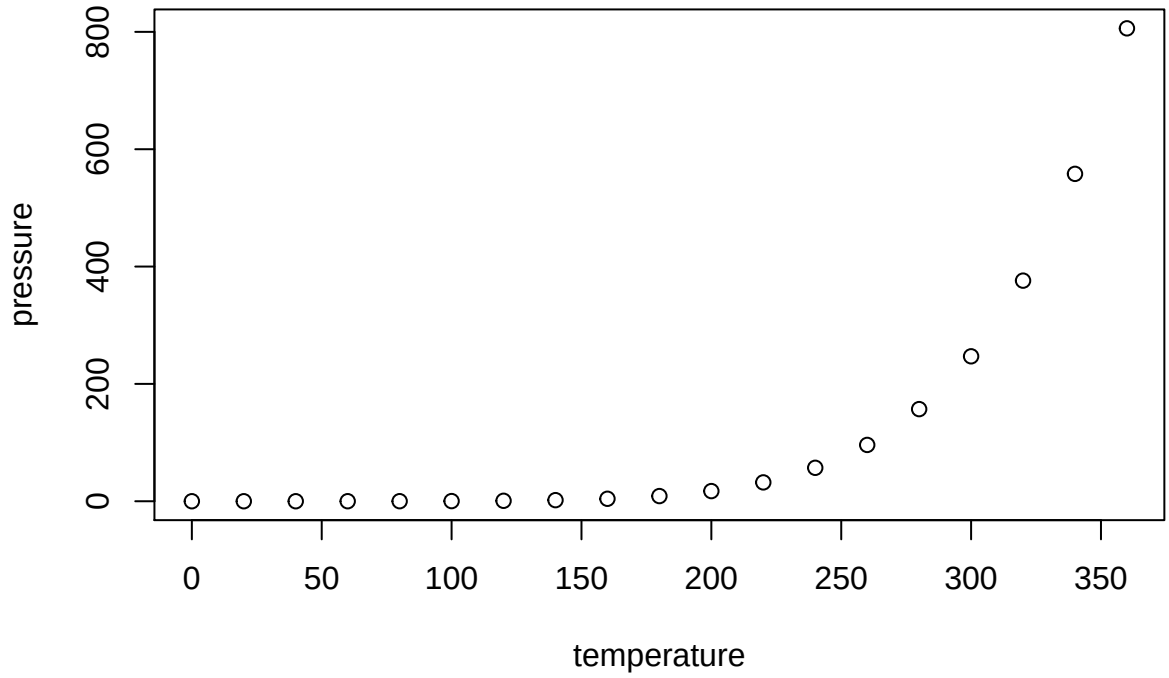
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00
```

Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

B202280 source code for assessment in R.

Link to my github repository

https://github.com/B202280/B202280_assessment

Loading NHS datasets

library(NHSRdatasets) library(tidyverse) library(here) library(knitr) library(scales) library(lubridate) library(caret) #Load the ae_attendances data. data(ae_attendances)

ae<-ae_attendances class(ae)

Viewing the data

ae

ae <- rowid_to_column(ae, "index")

ae %>% # Set the period column to show in month-year format mutate_at(vars(period), format, "%b-%y") %>% # Set the numeric columns to have a comma at the 1000's place mutate_at(vars(attendances, breaches, admissions), comma) %>% # Show the first 10 rows head(10) %>% # Format as a table kable()

I saved my data here

write_csv(ae, here("RawData", "ae_attendances.csv"))

Subsetting the data: I chose the following variables in order to focus on a+e attendance rates and how these may vary over time.

ae<-ae %>% select(index, period, attendances)

ae %>% # set the period column to show in Month-Year format mutate_at(vars(period), format, "%b-%y") %>% # set the numeric columns to have a comma at the 1000's place mutate_at(vars(attendances), comma) %>% # show the first 10 rows head(10) %>% # format as a table kable()

the glimpse function give us a snapshot of the data.

glimpse(ae) write_csv(ae, here("RawData", "ae_attendances_ENG.csv"))

#Now work out the proportion (`prop`) of the raw data to assign to the training data: prop<-(1-(15/nrow(ae))) #The proportion of the raw that needs to be assigned to the training data to ensure there is only 10 to 15 records in the test data is: print(prop)

#This will make sure that every time I run this script, I will partition the raw data into the same test and training data. set.seed(333) #Partitioning the raw data into the test and training data. trainIndex <- createDataPartition(ae$index, p = prop, list = FALSE, times = 1) head(trainIndex) # All records that are in the trainIndex are assigned to the training data. aeTrain <- ae[trainIndex,] nrow(aeTrain) #There are 12,753 records in my training data. That is a large dataset!

Now I will tabulate ae_attendances_ENG training data for my report

aeTrain %>% # set the period column to show in Month-Year format mutate_at(vars(period), format, "%b-%y") %>% # set the numeric columns to have a comma at the 1000's place mutate_at(vars(attendances), comma) %>% # show the first 10 rows head(10) %>% # format as a table kable()

And now save it to the Data folder.

write_csv(aeTrain, here("Data", "ae_attendances_ENG_train.csv"))

Extract the ae_attendances_ENG test data

#All records that are not in the trainIndex (`-trainIndex`) are assigned to the test data. aeTest <- ae[-trainIndex,] nrow(aeTest)

#Set aside the first record from the ae_attendances_ENG test data so that #I can test and evaluate my data-capture tool. aeTestMarker <- aeTest[1,]

Now tabulate ae_attendances_ENG marker test data for my report aeTestMarker %>% # set the period column to show in Month-Year format mutate_at(vars(period), format, "%b-%y") %>% # set the numeric columns to have a comma at the 1000's place mutate_at(vars(attendances), comma) %>% # show the first 10 rows head(10) %>% # format as a table kable()

Now to save my ae_attendances_ENG marker test data to my working data folder 'Data'

write_csv(aeTestMarker, here("Data", "ae_attendances_ENG_test_marker.csv"))

Now set aside the remaining records for me to test (or collect) with my data-capture tool.

aeTest <- aeTest[2:nrow(aeTest),]

Now tabulate ae_attendances_ENG test data for my report aeTest %>% # set the period column to show in Month-Year format mutate_at(vars(period), format, "%b-%y") %>% # set the numeric columns to have a comma at the 1000's place mutate_at(vars(attendances), comma) %>% # show the first 10 rows head(10) %>% # format as a table kable()

Now save my ae_attendances_ENG test data to my working data folder 'Data'

write_csv(aeTest, here("Data", "ae_attendances_test.csv"))

Data capture tool

Due to an error when I tried to load the panda package, I was unable to proceed to create my data capture tool in python.

Data dictionary for test data

library(dataMeta) library (tidyverse)

library(here)

CollectedData=read_csv(here("RawData", "CollectedDataAll.csv"))

Error: '/home/jovyan/B202280/Working_with_data_types_and_structure does not exist.

this error arose because I was not able to create my data capture tool in python.